Complexidade e Busca

NP, coNP, ..., PSPACE

Marcelo Finger

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

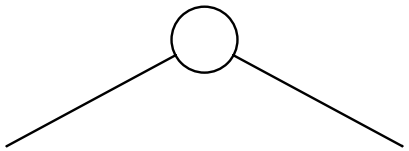
1º Semestre 2020

Tópicos

Tópicos

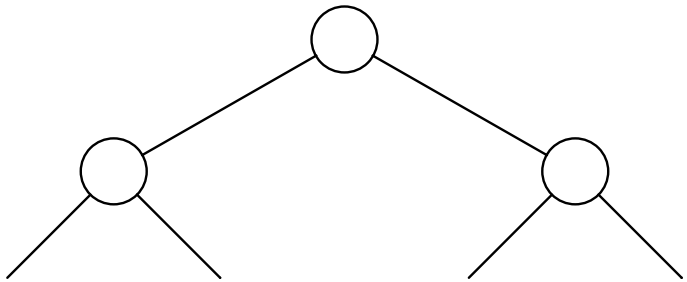
Encontrar um Estado Meta

Espaço de busca exponencial



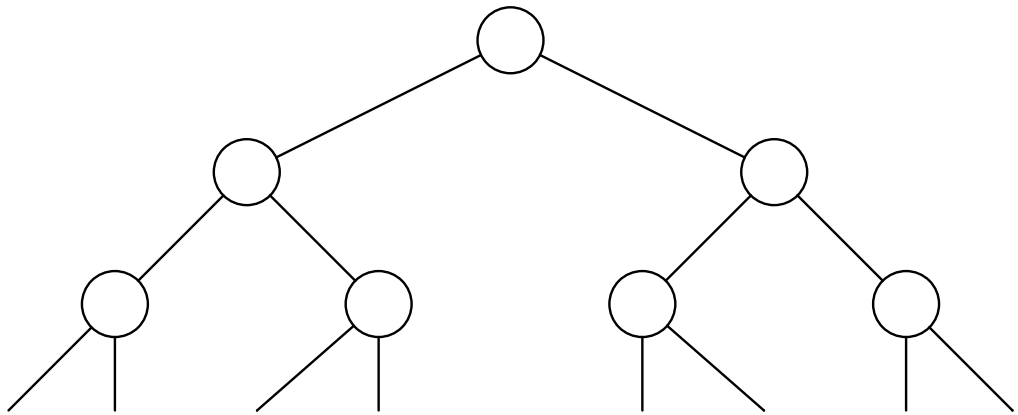
Encontrar um Estado Meta

Espaço de busca exponencial



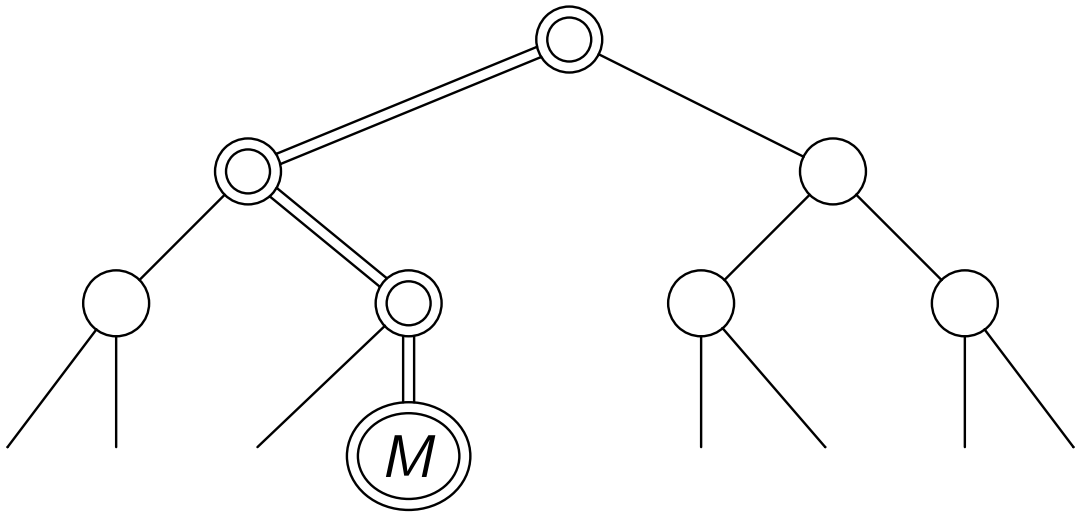
Encontrar um Estado Meta

Espaço de busca exponencial



Encontrar um Estado Meta

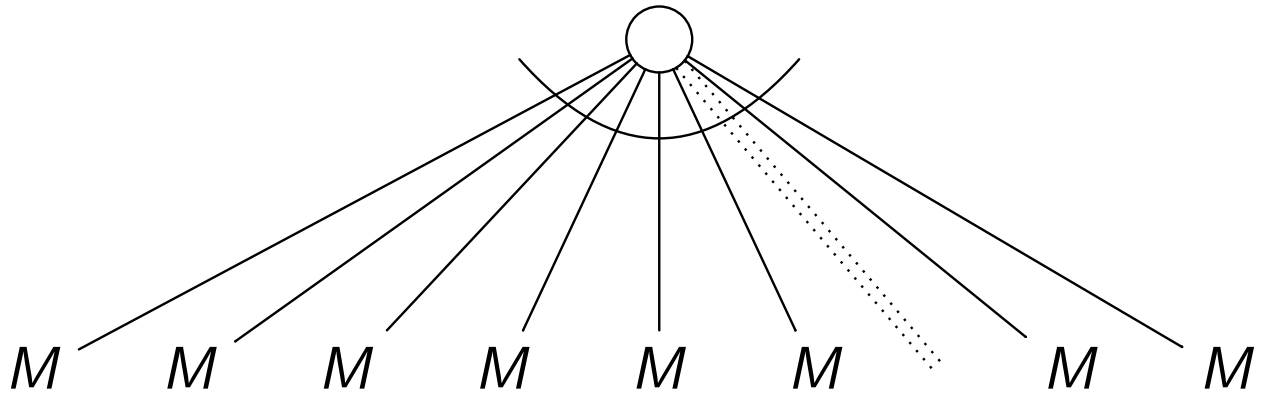
Espaço de busca exponencial



Caminho "testemunha" é pequeno: problema em NP (Σ_1^p)

Encontrar **Todos** Estados Meta

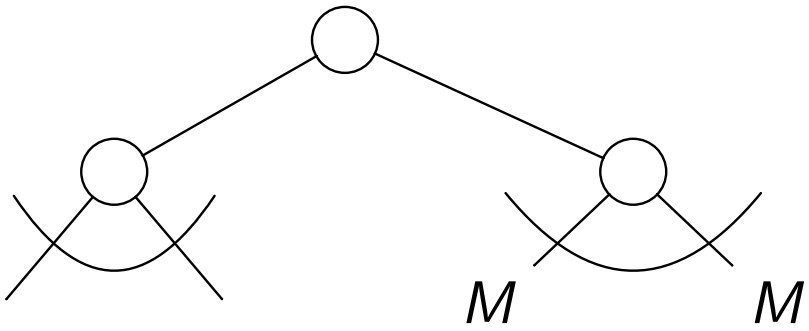
Espaço de busca igualmente exponencial



Contraexemplo "testemunha" é pequeno: problema em coNP (Π_1^p).

Alternância OU-E

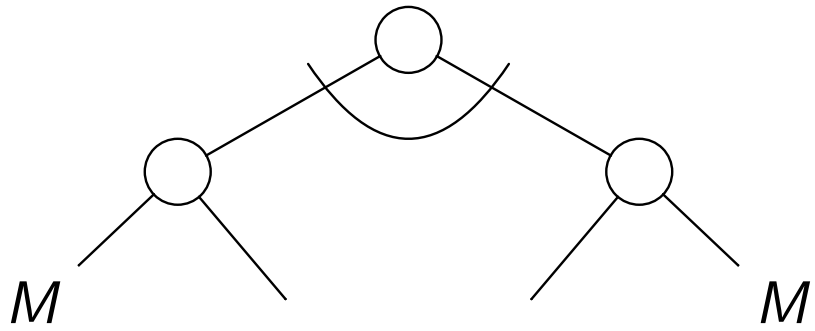
Espaço de busca exponencial



Encontrar caminho "testemunha" em que todos ramos são metas: problema em Σ_2^p

Alternância E-OU

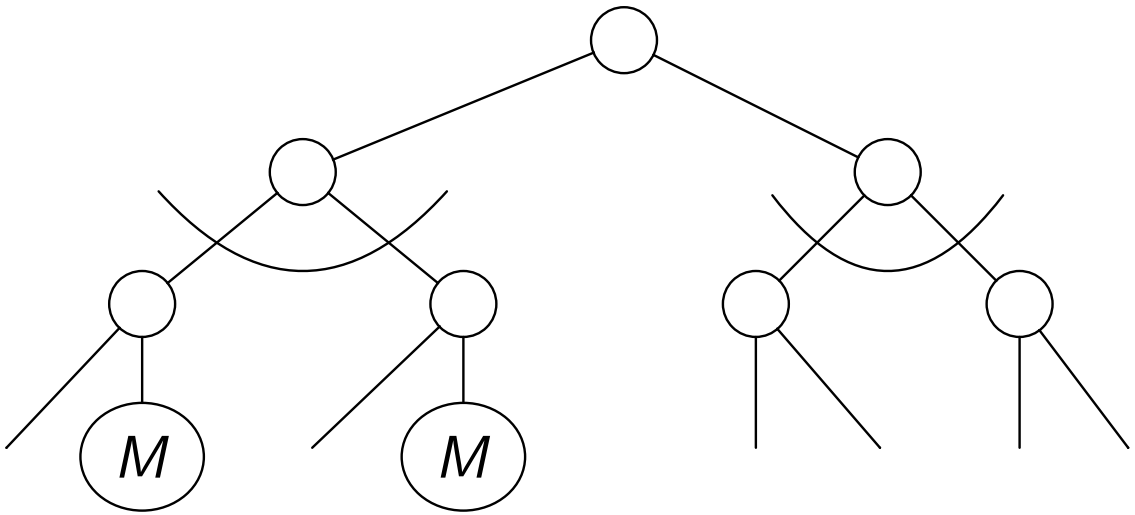
Espaço de busca exponencial



Encontrar todos os ramos em que há uma "testemunha" metas: problema em co$\Sigma_2^p = \Pi_2^p$

Alternância OU-E-OU

Espaço de busca exponencial



Existe caminho em que todas sub-árvores possuem um caminho para meta: Σ_3^p

Tópicos

Hierarquia de classes de complexidade

$$\Sigma_0^p = \Pi_0^p = P$$

$$\Sigma_{i+1}^p = \exists^p \Pi_i^p$$

$$\Pi_{i+1}^p = \forall^p \Sigma_i^p = \mathrm{co}\Sigma_{i+1}^p$$

Propriedares

$$\Sigma_i^p \subseteq \Sigma_{i+1}^p$$
$$\Sigma_i^p \subseteq \Pi_{i+1}^p$$

$$\Pi_i^p \subseteq \Pi_{i+1}^p$$
$$\Pi_i^p \subseteq \Sigma_{i+1}^p$$

Em aberto: Colapso da hierarquia polinomial

Existe algum i tal que $\Sigma_i^p = \Pi_i^p$?

Tópicos

Lógica Booleana

Temos um conjunto de símbolos/variáveis proposicionais:
$X = \{x_1, \ldots, x_n\}$

Conectivos Booleanos: $\neg, \wedge, \vee, \rightarrow$
Fórmulas construídas com esses conectivos

Semântica: valorações $v : X \rightarrow \{0, 1\}$
Dados v e fórmula φ, calcular $v(\varphi)$ é fácil

Satisfatibilidade e Validade

Dada uma fórmula φ

- φ é satisfazível se existe v, $v(\varphi) = 1$ (NP)
- φ é válida se para todo v, $v(\varphi) = 1$ (coNP)

Compare com

- $\exists x_1 \ldots \exists x_n \varphi$ é válida (NP)
- $\forall x_1 \ldots \forall x_n \varphi$ é válida (coNP)

Fórmulas da Lógica Booleana Quantificada (LBQ)

Fórmulas da LBQ

Seja φ uma fórmula booleana.
Então as fórmulas da LBQ tem a forma:

$$\exists x_1 \ldots x_{a_1} \forall x_{a_1+1} \ldots x_{a_2} \ldots Q x_{a_{m-1}+1} \ldots x_{a_m} \varphi$$

ou

$$\forall x_1 \ldots x_{a_1} \exists x_{a_1+1} \ldots x_{a_2} \ldots Q x_{a_{m-1}+1} \ldots x_{a_m} \varphi$$

m é o grau de alternância da fórmula.
$Q \in \{\exists, \forall\}$.

Complexidade

- Decidir a validade de uma fórmula LBQ é PSPACE-completo.
- Se fixarmos o grau de alternância, obtemos a Hierarquia Polinomial

Equivalência entre LBQ e BQ

Theorem

Para cada fórmula A da LBQ existe uma fórmula B sem quantificadores tal que A é valida sse B é satisfatível.

- B em geral é exponencialmente maior que A
- A transformação de B em A é o processo de **determinização** de uma busca não determinística.

Exemplos

- $\exists x\varphi$ é válido se φ é sat sse $(x \wedge \varphi) \vee (\neg x \wedge \varphi)$ é sat
- $\forall x\varphi$ é válido se $(x \rightarrow \varphi) \wedge (\neg x \rightarrow \varphi)$ é sat
- A tradução é feita algoritmicamente:

$$bool(\exists xA) = (x \wedge bool(A)) \vee (\neg x \wedge bool(A))$$

$$bool(\forall xA) = (x \rightarrow bool(A)) \wedge (\neg x \rightarrow bool(A))$$

Como fica esta tradução em termos de árvores E-OU?